EE244 Project

Zexin Li(862396120)*, Yufei Li(862396528)*

November 2022

1 Introduction

In recent years, academics and industry have paid close attention to dialogue generation (DG) models. DG model design and training are difficult to work with that require a lot of computer power, but they could have a significant impact in the actual world, such as assisting people in their daily lives. Real-world applications (such as emergency question-answering requirements, etc.) depend on real-time feedback. The ability of DG models to respond in these application contexts is essential. But current DG strategies mainly concentrate on enhancing model accuracy (also defending the adversarial accuracy-based attacks). The question of whether the DG model could continue to function effectively merits further research.

To evaluate the resilience of DG model efficiency, we must first determine the variables that would influence DG model efficiency. In this study, we explore a characteristic of DG models: The Markov Process used by the DG model to generate output tokens makes the underlying decoder calls non-deterministic. As a result, DG models' computational requirements are by nature non-deterministic. This innate characteristic reveals a possible weakness in DG models. As a result, adversaries might be able to create specialized adversarial inputs that significantly raise the computing cost of DG models. In real-world circumstances, this efficiency vulnerability could have serious consequences. Efficiency-based attacks, for instance, may result in significant, redundant processing resources and negatively impact the user experience by raising device battery consumption or prolonging the response latency. In this project, we plan to investigate such potential vulnerability by answering the following questions: Can we add adversarial perturbation modifications to text inputs to significantly increase the computational consumption of DG models and degrade the model efficiency? If so, how severe the efficiency degradation can be?

In this project, we propose a new methodology, DGSlowdown, to generate efficiency-oriented adversarial inputs. To be specific, DGSlowdown will apply the minimal perturbation on the benign inputs that could minimize the likelihood of an End Of Sentence (EOS) token and delay the appearance of EOS accordingly. We evaluate DGSlowdown on four subject models with two popular datasets and compare with six baselines. The evaluation results show that it's necessary to improve and protect the efficiency and robustness of DG models.

2 Related Work

2.1 Adversarial Attack

Adversarial examples commonly exist in various neural networks, which is explored as an intriguing property of the neural network. [9] Such vulnerability is first exploited by the gradient-based adversarial attacks I-FGSM [4] to spoof image classification models. Several mainstream adversarial machine learning methods have improved the optimization methods. For instance, MI-FGSM [2] incorporates momentum with gradient information and introduces ensemble models to craft stronger adversarial examples. VMI-FGSM [10] further improves upon the MI-FGSM on the attacking performance by alleviating the gradient variance with a sampling method. In natural language processing, researchers treat the adversarial attack as a harder searching problem since the discrete input nature limits the direct use of gradient-based attacks. For instance, PWWS [7] performs effective word-level attacks against text classification by greedily substituting words. WSLS [12] boosts the attacking performance by speeding up local search by word saliency.

Unlike previous work that focuses on adversarial attacks for accuracy goals, we give a new angle at this problem generation adversarial attacks making the system energy consumption rising by extending the output length, which may significantly decrease the efficiency of the targeted system.

2.2 Energy Attack

Due to the high inference time cost of DNNs and resource-constraint facts of edge-embedded devices, the efficiency of DNNs has attracted wide attention. Hence, the study of robustness for network energy consumption has also received some attention. Recent research by Haque et al. [3] has shown that specific DNNs (e.g., input adaptive DNNs, etc.) are not robust against energy-goal attacks. In other words, the input adaptation designs could not save computational costs under energy-goal attacks. In addition, Chen et al. [1] proposes energy attacks against neural network-based image captioning systems with a gradient-based attacking method and significantly degrade the efficiency of such image captioning systems (i.e., the systems generate extremely long image captions under imperceptible adversarial perturbations). These works reveal the vulnerability of DNNs' efficiency and suggest that the system whose outputs are text sequences may be vulnerable to energy attacks.

Different from all existing works, we are the first to choose Dialogue Generation (DG) as the target to perform the energy attack. Our proposed method considers the challenge of attacking DG models (the input is discrete space. Hence it is difficult to adapt gradient-based attack) and adopt a search-based optimization algorithm to overcome such challenge.

3 Technical Approach

3.1 Dialogue Generation

In natural language processing, dialogue generation (as shown in Figure 1) is the process of "understanding" natural language inputs to produce output. The systems are typically designed to engage in human-to-human communication, such as a back-and-forth conversation with a conversation agent or a chatbot. FusedChat and the Ubuntu Dialogue Corpus are two examples of benchmarks for this task (see others like Natural Language Understanding). Metrics like BLEU, ROUGE, and METEOR can be used to evaluate models, but they have limitations due to their poor correlation with human judgment. New metrics like unsupervised and reference-free (USR) and metrics for automatic Unreferenced dialog evaluation may overcome these limitations.

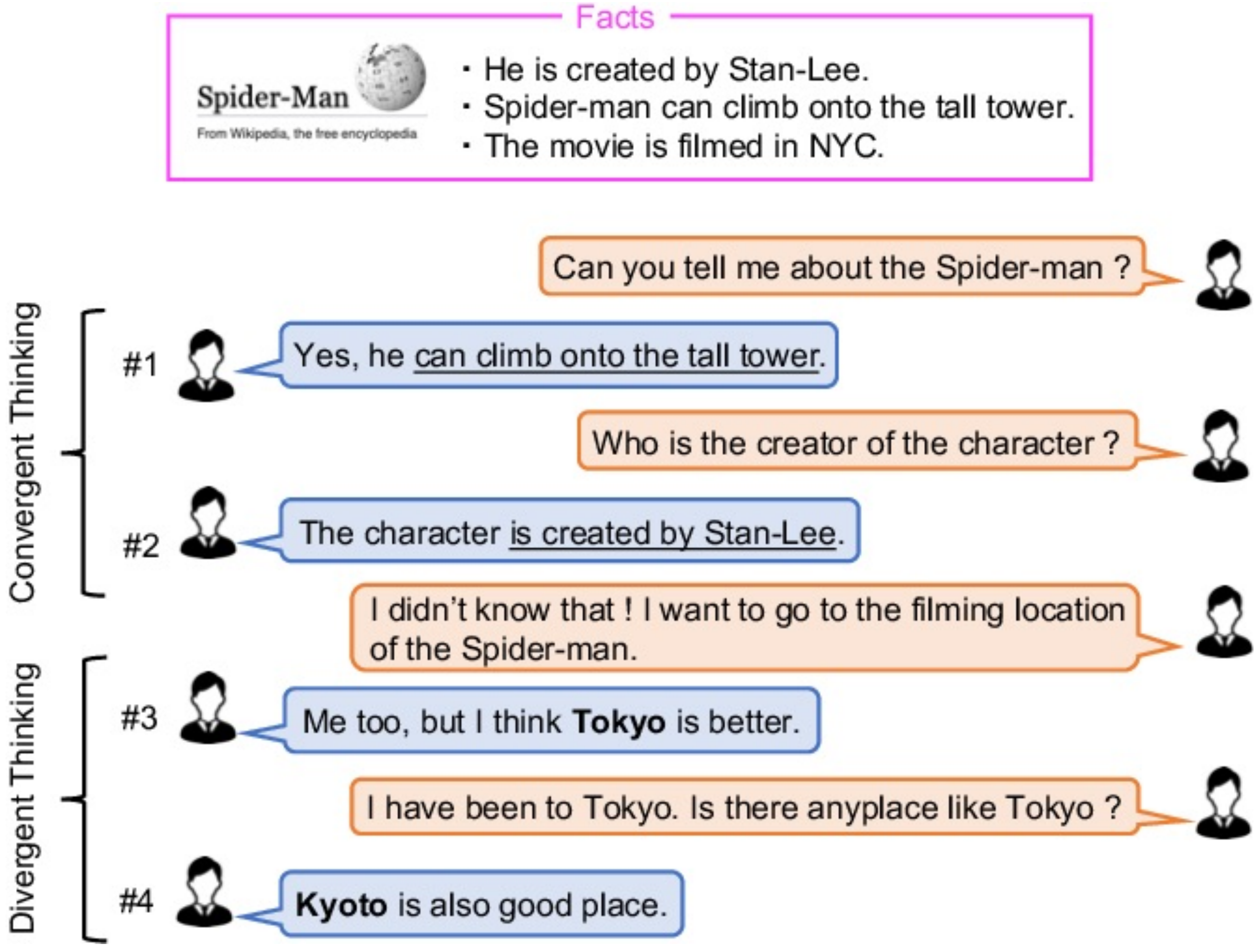


Figure 1: An example of Dialogue Generation (DG) model

3.2 Problem Formulation

$$\begin{aligned} \Delta &= \mathrm{argmax}_{\delta} \quad \mathrm{Loop}_{\mathcal{F}}(x+\delta) \\ s.t. & \quad ||\delta|| \leq \epsilon \ \wedge \ ||x+\delta|| \in [0,1]^n \end{aligned} \tag{1}$$

Our objective is to generate human-unnoticeable perturbations to input text to decrease the efficiency of the victim DG model during inference time. The goal focuses on three things: *(i)* The adversarial image produced should make the victim DG model more computationally demanding. *(ii)* humans are unable to distinguish between the created adversarial image x' and the positive image x; *(iii)* It is expected that the created adversarial image x' will be accurate in practice. We formulate the three factors described above in Eq.1. x is the benign input, $\mathcal{F}$ is the victim DG model under attack, ϵ is the maximum adversarial perturbation allowed, and $\text{Loop}_{\mathcal{F}}(\cdot)$ measures the number of decoder calls in the victim DG model $\mathcal{F}$. Our proposed approach DGSlowdown tries to optimize a perturbation Δ that maximizes the number of decoder calls while meeting the unnoticeable constraints of perturbation.

3.3 Attack Overview

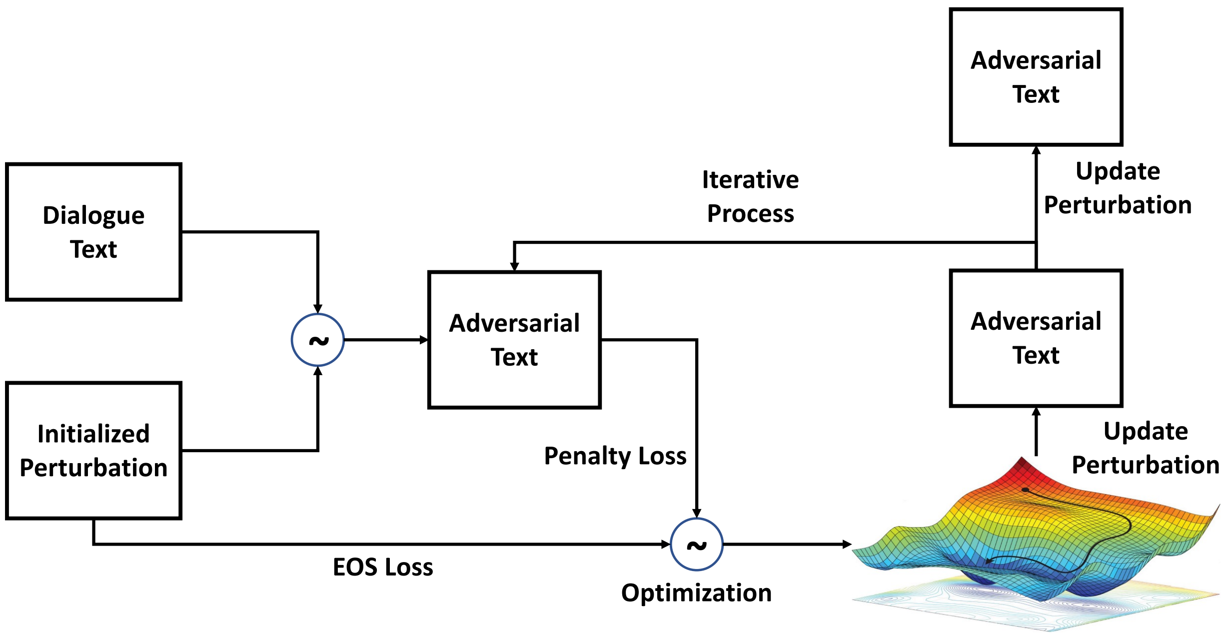


Figure 2: Workflow of DGSlowdown

Figure 2 shows the overview of our proposed attack. Given a benign input text, DGSlowdown first makes an adversarial perturbation with zero-initialization. After that, DGSlowdown computes the efficiency reduction loss (Eq. 2) and the perturbation penalty loss (Eq. 3) and adds them up in a multi-tasking manner (Eq. 4) The reduction loss aims to slow down the victim DG model, and the perturbation penalty loss seeks to enforce the generated adversarial examples to satisfy the unnoticeable constraints (Eq.1). Finally, DGSlowdown updates the adversarial perturbation by jointly optimizing the perturbation penalty loss and the efficiency reduction loss.

3.4 Energy-goal Targeted Mechanism

We propose Energy-goal Targeted Mechanism(EGTM) to guide the generation of compelling adversarial examples. The core goal of this mechanism is to prevent the age of a specific output token (EOS token) in an adversarial way, thus achieving the energy goal of the force model to generate a very long output.

Specifically, the raw text of the discrete states is tokenized by the tokenizer action. Each token is input to the Dialogue Generation model for inference, resulting in an output embedding generated by the corresponding detokenizer action. (See section 3.1.1.) The output is terminated if the token generated by the detokenizer is an EOS token.

$$\mathcal{L}_{eos} = \frac{1}{n}\sum_{i=1}^{n}\left\{l_i^{eos} - \mathbb{E}_{k\sim p_i} l_i^k\right\} \tag{2}$$

Intuitively, suppose we do not consider the Markov relationship implicit in the Dialogue Generation (DG) model. In that case, we only need to force the adversarial sample to reduce the probability of outputting an EOS token, which can serve as an energy attack. We apply minimum likelihood estimation (MLE) to minimize the likelihood of EOS tokens, as shown in Eq.2. Furthermore, we use a penalty loss design as a regularizer to prevent overfitting, as demonstrated in Eq.3.

$$\mathcal{L}_{per} = \begin{cases} 0; & \text{if } \delta \le \epsilon \\ ||\delta - \epsilon||; & \text{otherwise} \end{cases} \tag{3}$$

Our final loss can be formulated as a weighted sum of EOS loss and penalty loss, shown as Eq. 4, where λ is a hyper-parameter that controls the penalty term's impact level.

$$\mathcal{L} = \mathcal{L}_{eos} + \lambda\mathcal{L}_{per} \tag{4}$$

3.5 Search-based Attacking Strategy

Another challenge in performing an energy attack on the DG model is that the input space is discrete. Most implementations of adversarial attacks in computer vision use gradient-based methods because they default to semantically continuous inputs (pixel value from [0,255]) and can be solved optimally using gradient descent methods. However, for two semantically similar tokens, although the distance on the embedding may be closer, the difference in the values corresponding to the tokens may be huge because the values corresponding to the tickets are related to the dictionary used by the tokenizer.

$$fit(x) = length(\mathcal{F}(x)) \tag{5}$$

To address this challenge, we propose a Search-based Attacking Strategy(SBAS) tailored to Dialogue Generation(DG) models. Firstly, we leverage the gradients for the embedding; we perform word-level attacks by masking each word correspondingly and use K-NN to generate K candidates input with the smallest distance from the embedding for each mask. Then, for each epoch, generated candidates are filtered by the fitness function (Eq. 5), those most adapted samples are selected, and others are discarded. Here, we use the output length

Algorithm 1 DGSlowdown Attack

Input: Benign input x
Input: Victim DG model $\mathcal{F}(\cdot)$
Input: Maximum perturbation ϵ
Input: Maximum Iterations T
Output: Adversarial examples x' that satisfy Eq.1

1: $\delta \Leftarrow 0$ Initialize δ with 0. Initialize x_{adv} with x.
2: **for** *iter* in Range(T) **do**
3:     $S_{x_{adv}} = \emptyset$
4:     $\mathcal{L}_{eos} = L_1(x_{adv}, \mathcal{F})$ Compute $\mathcal{L}_{eos}$ by Eq.2.
5:     $\mathcal{L}_{per} = L_2(\delta, \epsilon)$ Compute $\mathcal{L}_{per}$ by Eq.3.
6:     $\mathcal{L}_{total} = \mathcal{L}_{deg} + \lambda \mathcal{L}_{per}$ Compute joint loss by Eq.4.
7:     $\nabla = \frac{\partial \mathcal{L}_{total}}{\partial w}$ Compute the gradients
8:     **for** *token* in x **do**
9:         $S_{x_{adv}} = S_{x_{adv}} \cup Mutation(x, \nabla)$
10:     **end for**
11:     $x_{adv} = Selection(S_{x_{adv}}, fit)$ Compute fitness function by Eq.5
12: **end for**
13: Return x_{adv} Return the adversarial example.

of the outputs of the victim model as the fitness function. By adopting such a method, we can efficiently generate strong adversarial examples.

3.6 Attacking Algorithm

The detailed algorithm is shown as Algorithm 1. Our attack algorithm accepts four inputs: a benign input image x, the victim DG model $\mathcal{F}$, a pre-defined perturbation threshold ϵ, and the maximum iteration number T. Our algorithm outputs an adversarial example x_{adv} that satisfy Eq.(1).

Our algorithm first initializes the adversarial perturbation as zero (line 1). After that, we iteratively update the adversarial examples x_{adv}. Specifically, in each iteration, we compute the EOS loss based on Eq.2 and the perturbation penalty loss based on Eq.3. We then optimize w by minimizing the weighted joint losses on Eq.4 and get the adversarial gradient ∇. Next, we leverage this gradient to generate mutations for masking each token. Then we use the fitness function to keep the best-fit examples greedily.

4 Experimental Results

4.1 Experimental Setup

4.1.1 Datasets

Persona for Unguided Speaker:	**Persona for Guided Speaker**:
My son plays on the local football team.	My eyes are green.
I design video games for a living.	I wear glasses that are cateye.

Wizard of Wikipedia topic: Video game design
Previous utterances (shown to speakers):
U: What video games do you like to play?
G: all kinds, action, adventure, shooter, platformer, rpg, etc. but video game design requires both artistic and technical competence AND writing skills. that is one part many people forget

Actual utterances:
U: Exactly! I think many people fail to notice how beautiful the art of video games can be. (ConvAI2)
(G selected the WoW suggestion: "Indeed, Some games games are purposely designed to be a work of a persons creative expression, many though have been challenged as works of art by some critics.")
G: Indeed, Some games games are purposely designed to be a work of a persons creative expression, many though have been challenged as works of art by some critics. (WoW)
U: Video games are undervalued by many and too easily blamed for problems like obesity or violence in kids (WoW)
G: Indeed, Just last week my son was playing some Tine 2 and it was keeping him so calm.
Games are therapeutic to some. (ED)
U: I use games to relax after a stressful day, the small escape is relaxing. (ConvAI2/ED)
(G selected the ED suggestion: "I enjoy doing that after a hard day at work as well. I hope it relaxes you!")
G: I enjoy a good gaming session after a hard day at work as well. (ConvAI2/ED)
U: What other hobbies does your son have? (ConvAI2)
G: Well he likes to fly kites and collect bugs, typical hobbies for an 8 year old, lol. (ConvAI2)
U: My 12 year old is into sports. Football mostly. I however don;t enjoy watching him play. (ConvAI2)
G: I wish I could play football, But I wear this cateye glasses and they would break if I tried. (ConvAI2)
U: Sounds nice. Are they new or vintage? (ConvAI2)
G: They are new, I got them because of my love for cats lol. I have to show off my beautiful green eyes somehow. (ConvAI2)

Figure 3: Sample conversation from the Blended Skill Talk dataset. Individual utterances are annotated with the single-skill datasets they are reminiscent of. The conversation here has been seeded with two utterances.

We choose Blended Skill Talk [8] (BST) as the dialogue generation dataset, which contains 7k conversations explicitly designed to exhibit multiple conversation modes: displaying personality, having empathy, and demonstrating knowledge. Specifically, 76k utterances was collected with a guided and unguided human speaker, where the guided speaker could select utterances suggested by bots trained on the three individual tasks (see Figure 3).

In each blended dialogue, the model is provided a two-sentence persona to condition on following PersonaChat [11] (see Figure 3). During evaluations, we equip our models with randomly chosen personas and mirroring the way the model is trained.

4.1.2 Dialogue Generation Models

We consider a standard Seq2Seq Transformer architecture to generate responses rather than retrieving them from a fixed set. Our implementation is based on BART hugging face version [5]. BART is a denoising autoencoder that maps a corrupted document to the original document it was derived from. It is implemented as a sequence-to-sequence model with a bidirectional encoder over corrupted text and a left-to-right autoregressive decoder (see Figure 4). We use Byte-level BPE tokenization [6] pre-trained on open-domain datasets,

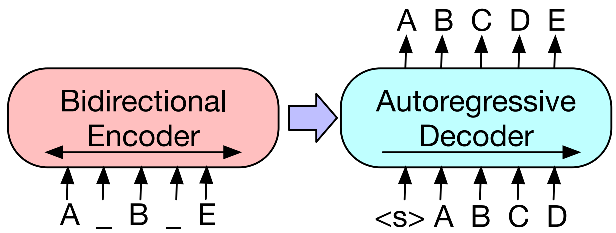


Figure 4: Architecture of seq2seq BART. The corrupted document (left) is encoded with a bidirectional encoder, and then the likelihood of the original document (right) is calculated with an autoregressive decoder.

as implemented in huggingface's Tokenizers. We consider BART-base – a model of 139M parameters. It has a 6-layer encoder, a 6-layer decoder with 768-dimensional embeddings, and 12 attention heads.

4.1.3 Metrics

Our evaluation considers both hardware-independent metrics (output length, BLEU) and hardware-dependent metrics (overhead), which quantitively represent the dialogue system's effectiveness and efficiency.

Output length. The generation output length measures the hardware-independent decoding efficiency. The longer output indicates that more computations and a larger response time are required to chat with users.

Latency. Response latency is a hardware-dependent metric that measures system efficiency. A larger latency indicates less efficient dialogue systems.

Bilingual evaluation understudy (BLEU). BLEU is a metric designed for evaluating the quality of machine-translated text from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human.

Attack success rate (ASR). A success attack is the adversarial input u' (generated from original input u) that induces longer output length.

4.1.4 Implementation Details

We first train a seq2seq BART on the training dataset of BST. Then we sample part of the test dataset of BST to implement our adversarial attack.

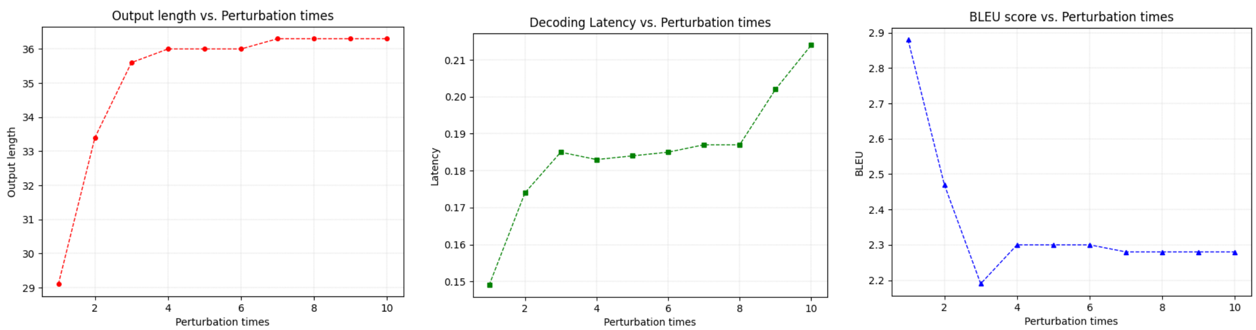
Training. We train a BART model on the training dataset of BST. For a pair of guided (g) and unguided (u) utterances, the model $f(\cdot)$ takes unguided utterance u as input and tries to approximate the guided response g. Cross entropy is calculated as the training loss $\mathcal{L} = CE(f(u), g)$.

We use validation BLEU as the criteria for early stopping to avoid overfitting. We train 50 epochs with batch size 16 and initial learning rate 1e-5. We use Adam as the optimizer.

Adversarial attack. We sample 100 pairs of (u, g) dialogues from the test dataset of BST. For each input u, we consider word-level perturbations and use the BCE loss with respect to the EOS token to search for tokens that maximize the embedding gradient. We

Method	ASR (%)	Output length	Latency (s)	BLEU
Original	-	15.67	0.09	2.87
Ours	100	33.13	0.17	2.36

Table 1: Evaluation results of the attack method on the BST test dataset.



(a) Output length (b) Latency (c) BLEU

Figure 5: Dialogue generation performance vs. perturbation times.

set the maximum attacking iteration as 3, meanings that each input u is attacked 3 times iteratively, and we select the best adversary u' that induces the longest output.

4.2 Quantitative Results

We calculate the average output length, overhead, and BLEU on the sampled BST test dataset (see Table 1). Our attack reaches a 100% attack success rate, meaning that each perturbation u' generates a longer output sequence than the original u. Specifically, the average output length and overhead are twice as the original results, proving that our attack effectively reduces the efficiency of the dialogue model. Lower BLEU indicates that our attack also degrades the quality of the dialogue generation from a linguistic perspective.

4.2.1 Visualization of Adversarial Examples

We select three pairs of sequence-to-sequence dialogue examples and the corresponding adversarial examples (with the generation outputs). We visualize these cases in Table 2, where we can see that by replacing only several tokens of the original sentence, the generated sequence becomes much longer than before.

4.2.2 Ablation Study

We show the relationship between perturbation times and the evaluation metric scores (see Figure 5). In general, more attack iterations induce longer output, higher latency, and smaller BLEU, proving the effectiveness of our attack.

Input sequence	Generated sequence (length)
Congrats also and hope he bring you all bundle of joy. Congrats abortions and hope squats bring you all bundle of joy.	Thank you! He will be a bundle of joy! (11) Yeah, I hope so too. I'm currently trying to get my meds in so I don't have to deal with all that stress! (28)
Great we have a future looking through the eyes of our children. Great we forbids a future looking through the eyes of ALL children.	That's great. I hope you have a bright future. (12) I think that's a good point. I think it's good that we're looking at the eyes of ALL children, not just children. (31)
Cheers! *clinks glasses* Cheinding! *clinks glasses bob	haha, so what do you do for fun? (14) hahaha I know what you're talking about. I'm a dental hygienist so I have to keep reminding myself of that. (33)
Please tell us the date. I know you guys are happy. Please tell us the platOPA iien you guys are happy	Tomorrow. (2) I'm happy for you guys. I'm not sure what the platOPA is yet, but I'm sure it will be fine. (27)
Thank you what joy the babies brings. Deducted joy what joy many babies brings.	They bring me so much joy! (8) They bring me so much joy! I don't get to see them much, but I do get to spend a lot of time with them. (29)

Table 2: Examples of original dialogue generation (blue) and the corresponding adversarial examples (red).

5 Conclusions

In this study, we aim to degrade the DG models' efficiency by introducing an energy-goal attack. The efficiency of DG models is inversely related to the length of DG output sequences, a possible vulnerability of DG models that we explore with the DGSlowDown algorithm that provides adversarial efficiency-reducing inputs. Based on the thorough study, we can see that DGSlowDown can produce inputs that significantly reduce the effectiveness of DG models. We believe this is the first adversarial assault to examine the effectiveness and robustness of DG models.

References

[1] Simin Chen, Cong Liu, Mirazul Haque, Zihe Song, and Wei Yang. Nmtsloth: understanding and testing efficiency degradation of neural machine translation systems. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1148–1160, 2022.

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9185–9193, 2018.

[3] Mirazul Haque, Anki Chauhan, Cong Liu, and Wei Yang. Ilfo: Adversarial attack on adaptive neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14264–14273, 2020.

[4] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

[6] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[7] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics.

[8] Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. Can you put it all together: Evaluating conversational agents' ability to blend skills, 2020.

[9] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[10] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1924–1933, 2021.

[11] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[12] Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1967–1977, Online, August 2021. Association for Computational Linguistics.